

WHAT IS CLAIMED IS:

1                   1.       A very long instruction word (VLIW) processing core comprising:  
2                   a processing pipeline having N-number of processing paths for processing an  
3 instruction comprising N-number or P-bit instructions appended together to form a VLIW,  
4 said N-number of processing paths process said N-number of P-bit instructions in parallel on  
5 M-bit data words; and

6                   one or more register files having Q-number of registers, said Q-number of  
7 registers being M-bits wide;

8                   wherein one of said Q-number of registers in at least one of said one or more  
9 register files is a program counter register which stores a current program counter value.

1                   2.       The processing core as recited in claim 1, wherein one of said Q-  
2 number of registers in at least one of said one or more register files is a zero register which  
3 always stores zero.

1                   3.       The processing core as recited in claim 1, wherein program jumps are  
2 executed by adding a value to the current program counter value stored in the program  
3 counter register using a standard add operation.

1                   4.       The processing core as recited in claim 1, wherein memory addresses  
2 are calculated by adding a value to the current program counter value stored in the program  
3 counter register using a standard add operation.

1                   5.       The processing core as recited in claim 1, wherein program jump tables  
2 hold values, which are offset values from the current program counter value.

1                   6.       The processor chip as recited in claim 1, wherein M=64, Q=64, and  
2 P=32.

1                   7.       The processing core as recited in claim 1, wherein said Q-number of  
2 registers within each of said one or more register files are either private or global registers,  
3 and wherein when a value is written to one of said Q-number of said registers which is a  
4 global register within one of said plurality of register files, said value is propagated to a  
5 corresponding global register in the other of said one or more register files, and wherein when  
6 a value is written to one of said Q-number of said registers which is a private register within

7 one of said one or more register files, said value is not propagated to a corresponding register  
8 in the other of said one or more register files.

1 8. The processing core as recited in claim 7, wherein  $Q=64$ , and a 64-bit  
2 special register stores bits indicating whether a register in a register file is a private register or  
3 a global register, each bit in the 64-bit special register corresponding to one of said registers  
4 in said register file.

1 9. The processing core as recited in claim 7, wherein said program  
2 counter register is a global register.

1 10. A processing core comprising:  
2 a processing pipeline having N-number of processing paths, each of said  
3 processing paths for processing instructions on M-bit data words; and  
4 one or more register files, each having Q-number of registers, said Q-number  
5 of registers being M-bits wide;  
6 wherein one of said Q-number of registers in at least one of said one or more  
7 register files is a program counter register which stores a current program counter value; and  
8 wherein said Q-number of registers within each of said one or more register  
9 files are either private or global registers, and wherein when a value is written to one of said  
10 Q-number of said registers which is a global register within one of said one or more register  
11 files, said value is propagated to a corresponding global register in the other of said one or  
12 more register files, and wherein when a value is written to one of said Q-number of said  
13 registers which is a private register within one of said one or more register files, said value is  
14 not propagated to a corresponding register in the other of said one or more register files.

1 11. The processing core as recited in claim 10, wherein one of said Q-  
2 number of registers in at least one of said one or more register files is a zero register which  
3 always stores zero.

1 12. The processing core as recited in claim 10, wherein program jumps are  
2 executed by adding a value to the current program counter value stored in the program  
3 counter register using a standard add operation.

1                   13.     The processing core as recited in claim 10, wherein memory addresses  
2     are calculated by adding a value to the current program counter value stored in the program  
3     counter register using a standard add operation.

1                   14.     The processing core as recited in claim 10, wherein program jump  
2     tables hold values, which are offset values from the current program counter value.

1                   15.     The processing core as recited in claim 10, wherein a processing  
2     instruction comprises N-number of P-bit instructions appended together to form a very long  
3     instruction word (VLIW), and said N-number of processing paths process N-number of P-bit  
4     instructions in parallel.

1                   16.     The processor chip as recited in claim 15, wherein M=64, Q=64, and  
2     P=32.

1                   17.     The processing core as recited in claim 16, wherein Q=64, and a 64-bit  
2     special register stores bits indicating whether a register in a register file is a private register or  
3     a global register, each bit in the 64-bit special register corresponding to one of said registers  
4     in said register file.

1                   18.     The processing core as recited in claim 10, wherein said program  
2     counter register is a global register.

1                   19.     In a computer system, a scalable computer processing architecture,  
2     comprising:  
3                   one or more processor chips, each comprising:  
4                   a processing core, including:  
5                   a processing pipeline having N-number of processing paths, each of said  
6     processing paths for processing instructions on M-bit data words; and  
7                   one or more register files, each having Q-number of registers, said Q-number  
8     of registers being M-bits wide;  
9                   an I/O link configured to communicate with other of said one or more  
10     processor chips or with I/O devices;  
11                  a communication controller in electrical communication with said processing  
12     core and said I/O link;

13                   said communication controller for controlling the exchange of data between a  
14 first one of said one or more processor chips and said other of said one or more processor  
15 chips;

16                   wherein said computer processing architecture can be scaled larger by  
17 connecting together two or more of said processor chips in parallel via said I/O links of said  
18 processor chips, so as to create multiple processing core pipelines which share data  
19 therebetween.

1                   20.     The computer processing architecture as recited in claim 19, wherein  
2 one of said Q-number of registers in at least one of said one or more register files is a zero  
3 register which always stores zero.

1                   21.     The computer processing architecture as recited in claim 19, wherein  
2 program jumps are executed by adding a value to the current program counter value stored in  
3 the program counter register using a standard add operation.

1                   22.     The processing core as recited in claim 19, wherein memory addresses  
2 are calculated by adding a value to the current program counter value stored in the program  
3 counter register using a standard add operation.

1                   23.     The computer processing architecture as recited in claim 19, wherein  
2 program jump tables hold values, which are offset values from the current program counter  
3 value.

1                   24.     The computer processing architecture as recited in claim 19, wherein a  
2 processing instruction comprises N-number of P-bit instructions appended together to form a  
3 very long instruction word (VLIW), and said N-number of processing paths process N-  
4 number of P-bit instructions in parallel.

1                   25.     The computer processing architecture as recited in claim 24, wherein  
2 M=64, Q=64, and P=32.

1                   26.     The computer processing architecture as recited in claim 19, wherein  
2 said Q-number of registers within each of said one or more register files are either private or  
3 global registers, and wherein when a value is written to one of said Q-number of said  
4 registers which is a global register within one of said plurality of register files, said value is

5 propagated to a corresponding global register in the other of said one or more register files,  
6 and wherein when a value is written to one of said Q-number of said registers which is a  
7 private register within one of said one or more register files, said value is not propagated to a  
8 corresponding register in the other of said one or more register files.

1 27. The computer processing architecture as recited in claim 26, wherein  
2 Q=64, and a 64-bit special register stores bits indicating whether a register in a register file is  
3 a private register or a global register, each bit in the 64-bit special register corresponding to  
4 one of said registers in said register file.

1 28. The computer processing architecture as recited in claim 26, wherein  
2 said program counter register is a global register.

1 29. In a processing core comprising a processing pipeline having N-  
2 number of processing paths, each of said processing paths for processing instructions on M-  
3 bit data words, and one or more register files having Q-number of registers, said Q-number of  
4 registers being M-bits wide, a method for jumping from one location in a program to another  
5 location in a program, comprising the steps of:

6 storing a current program counter value in a program counter register, which is  
7 one of said Q-number of register in at least one of said one or more register files; and

8 adding a value to said current program counter value stored in said program  
9 counter register using a standard add operation.

1 30. In a processing core comprising a processing pipeline having N-  
2 number of processing paths, each of said processing paths for processing instructions on M-  
3 bit data words, and one or more register files having Q-number of registers, said Q-number of  
4 registers being M-bits wide, a method for calculating a memory address, comprising the steps  
5 of:

6 storing a current program counter value in a program counter register which is  
7 one of said Q-number of register in at least one of said one or more register files; and

8 adding a value to said current program counter value stored in said program  
9 counter register using a standard add operation.